Teaching an Old Dog New Tricks: Collaboration Techniques for Legacy Data

Thalene T. Mallus
The National Institute of Mental Health, Bethesda, MD

ABSTRACT

The National Institutes of Health (NIH) is the Federal Government's primary facility for biomedical investigation. As with any large agency, the NIH has available to its investigators multiple resources for data management and analysis. Although many investigators now maintain and analyze their data using desktop software, many large legacy data sets are still actively analyzed on mainframe systems such as MVS.

The prospect of sharing or accessing legacy data cross-platform can often seem daunting to investigators faced with this challenge. This paper reviews some existing methods for sharing MVS legacy data , as well as several World Wide Web (WWW) solutions currently being implemented by SAS Institute, the NIH Computer Center, and Netscape Communications Corporation.

INTRODUCTION

The National Institutes of Health (NIH) is the Federal Government's primary facility for biomedical investigation. As the nation's largest medical research center, the NIH encourages collaborative efforts among scientists on campus, as well as among university investigators and collaborators from other countries.

The NIH makes available to its investigators numerous computer platforms and software packages to manage and analyze data. On the one hand, having a variety of platforms and analysis packages to choose from is advantageous since investigators can freely choose their preferred operating system and statistics package without having to invest time learning a new platform and programming language.

On the other hand, when asked to share their data with colleagues from other institutes or universities, collaborations can become tricky when colleagues are utilizing different analysis programs on other operating systems.

It is estimated that approximately 80% of data at the NIH is still maintained and analyzed on legacy systems such as MVS[1]. The type of legacy data ranges from financial/administrative information to scientific research data. These data often exist in a variety of file formats (e.g. as raw text files, package specific system files like SAS data sets or SPSS system files, or relational database management systems (RDMS)).

Since much MVS scientific data is usually in the form of raw text or package specific system files, this paper will review some existing methods for sharing this type of legacy data , as well as several World Wide Web (WWW) solutions currently being implemented by SAS Institute, the NIH Computer Center, and Netscape Communications Corporation.

COLLABORATION CONSIDERATIONS

When faced with the prospect of sharing MVS legacy data with colleagues, it is often useful to consider the following issues *prior* to choosing a data sharing method:

- Does the collaborator have access privileges to the MVS system (e.g. will the sharing take place on your system or is a transfer necessary)?
- If the data must be transferred to another site, what is the current format of the data and on what operating system will the data be received?
- Depending on the existing file format, will variable names and associated formats need to be re-established at the receiving site?

DATA TRANSFER CONSIDERATIONS

Calvert and Ma (1996) have noted that data transfers require careful attention to the unique technical circumstances of the data exchange. They also recommend that collaborators spend a bit of time teasing out the answers to the following questions before choosing a data transfer method.

- Is the data transfer a one time event?
- Is the transfer a single file or many files?
- Does the receiving site need all of the data or just a specific subset of the data?
- Will there be repeated or regularly scheduled transfers of files that are updated frequently?
- What type of platform and software does the receiving site have available?
- Who will handle the transfer and receiving of the files (e.g. how knowledgeable are the sending/ receiving personnel)?

DATA TRANSFER PROBLEMS

Anytime data is moved across platforms from one environment to another, the potential for problems exists. Calvert and Ma (1996) discuss three main technical problems that can arise when attempting to transfer data across platforms:

- Lack of Compatibility - Data cannot be accessed by the target computer and the transfer is aborted until the compatibility problem is solved.
- Conversion Failures- The target system is able to read the data, but does so imperfectly.
- Data Corruption-The transfer methods are unreliable, yielding hidden data anomalies.

Fortunately, the SAS® System is well equipped to deal with most of the above data transfer issues. Of course, it is always a good idea to use SAS quality assurance techniques, such as PROC CONTENTS, PROC MEANS, or PROC COMPARE to verify that the data received at the collaborating site is the same as the original data (Mallus, 1996).

SAS FORMAT SOLUTIONS

For those circumstances when it is necessary to access or share MVS legacy data, the ease with which this can be done depends on the type of data to be transferred.

Data usually consists of two parts, microdata and metadata (Currall, 1995). Microdata is essentially the raw numbers or text. An example of microdata would be the following:

```
  18 2 27 81 53 2  1
  25 2 31 80 49 2 41
  35 2 30 80 50 2  1
 692 1 26 80    1 96
  49 2 26 83 56 1 96
etc....
```

Metadata is generally referred to as “data about the data”. In the example above there are seven columns of data, each representing a variable in the data set. The first column of numbers represents the SUBJECT ID, the second column represents GENDER, the third column represents YRBORN, (the last 2 digits of the year the subject was born) etc.

Further, variables such as GENDER usually have format values associated with the data, such as 1=male, 2=female. For large or complicated data sets it is generally preferable to transfer the metadata along with the microdata so the associated information does not have to be re-established at the receiving site (Currall, 1995).

Generally, transferring microdata across platforms is not much of a problem, since most operating systems can receive and read files in text format. However, the ease with which metadata can be sent along with the microdata can be somewhat of a challenge. Fortunately the SAS System offers a simple solution.

SAS TRANSPORT FILES

It is not possible to simply transfer a permanent SAS data set from one operating system to another. Rather, the data must be converted into a format that the receiving operating system can read (Kuligowski & Roberts, 1997).

SAS transport files (portable files) are sequential files which are independent of the host operating system and usually contain both micro and metadata. These files can be transferred electronically or on a physical media to the receiving operating system The process involves three simple steps:

1) Export - create the transport file on the original operating system.
2) Transport - move the file electronically, or write to a tape or floppy media.
3) Import - read the file back into SAS with the format of the receiving operating system.
(Kuligowski & Roberts, 1997)

An example program which converts a MVS SAS system file to a SAS transport file is:

```
//  (JOB statement)
//  JCLLIB ORDER=(ZABCRUN.PROCLIB)
//  EXEC DSSCR,NAME='aaaaiii.dsname2’
//  EXEC SAS
//SYSIN DD *
libname IN 'aaaaiii.dsname1' DISP=SHR;
libname OUT XPORT 'aaaaiii.dsname2'
      DISP=(NEW,CATLG) UNIT=FILE
      SPACE=(TRK,(s1,s2),RLSE)
      RECFM=FB LRECL=80 BLKSIZE=8000;
proc copy in=IN out=OUT;
title ‘Creating a SAS Transport File
      for MVS Export’;
endsas;
```

Once the transport file is relocated to the receiving host system, a similar program is run on that operating system to import the data (see SAS Technical Report P-195, Transporting SAS Files between Host Systems, for specific platform information).

INTERFACE LIBRARY ENGINES

As is often the case at NIH, collaborations are frequently complicated by the fact that the data to be shared is in another vendor’s file format (e.g. SPSS or BMDP system files).

With the introduction of Version 6 of the SAS System, a very useful set of internal routines called engines became available. SAS uses these engines to access a variety of data formats. This concept is known as Multiple Engine Architecture™. (For a complete discussion of SAS's Multiple Engine Architecture, the reader should refer to the SAS Language Reference, Version 6, First Edition.)

Briefly, there are two types of SAS library engines, *native* and *interface.* Native library engines access data sets saved in a variety of SAS formats, such as the default MVS Version 6 format (V609), the older SAS Version 5 format (V5), the SAS transport engine (XPORT), as well as several others.

The interface library engines are routines that allow SAS to directly read other vendor's file formats. SAS is able to access BMDP, SPSS, and OSIRIS files without prior conversion of the data to text or portable format.
The syntax for the SPSS interface library engine is as follows:

LIBNAME libref **SPSS** 'filename'**;**

Libref refers to the SAS libref, SPSS is the keyword for the SPSS library engine, and filename refers to the SPSS physical file name.

A MVS program example of the SPSS interface library engine is as follows:

```
//  (JOB statement)
//  JCLLIB ORDER=(ZABCRUN.PROCLIB)
//  EXEC SAS
//SYSIN DD *
libname IN SPSS 'aaaaiii.spssname';
proc contents data=IN._FIRST_;
title ‘Accessing a SPSS System File
      with SAS’;
endsas;
```

(For detailed information on the three interface library engines, please refer to the SAS Companion for the MVS Environment, Version 6, Second Edition.)

The advantage of utilizing SAS interface library engines is that both the micro and metadata can be accessed directly from the software specific system file (e.g. BMDP, SPSS, OSIRIS files) without any prior conversion of the data. This solution is particularly effective when collaborators on the same system simply need to read or perform SAS procedures on data which exist in another vendor's format. This avoids the additional steps of transporting or converting the data out of another vendor's internal format to SAS format or ASCII text for processing.

Output 1 in the Appendix provides an example of the results of a SPSS permanent file being read by the SAS system. Since SPSS files only have one logical member per file, _FIRST_ was used to access the data from the CONTENTS procedure.

Note that the CONTENTS procedure is a useful method to employ to orient to unfamiliar data as it provides information such as the file creation date, the file engine and vendor version number, the number of observations, variables and their attributes (if previously defined in the SPSS system file).

PROC CONVERT

A similar collaboration problem which is readily solved by the SAS System is the circumstance where a coworker's data is in another vendor's format on the same system and you wish to convert it to a SAS data set in order to simplify data management.

PROC CONVERT can be used to transform BMDP, SPSS, and OSIRIS files to SAS data sets. The following code example illustrates the conversion of a MVS SPSS system file to a temporary SAS data set.

```
//  (JOB statement)
//  JCLLIB ORDER=(ZABCRUN.PROCLIB)
//  EXEC SAS
//SYSIN DD *
filename  IN 'aaaaiii.spssname';
proc convert SPSS=IN out=TEMP;
title 'Converting SPSS Data to SAS
       Data';
proc contents;
endsas;
```

The example above produces a temporary SAS data set (TEMP) that includes all of the cases, variables and attributes of the original SPSS system file.

SAS CROSS-PLATFORM SOLUTIONS

The SAS System offers several types of data access tools in a client-server environment. SAS/CONNECT® and SAS/SHARE® software can be particularly useful collaboration tools when one is trying to optimize system or office resources.

SAS/CONNECT software is a data access tool which is used to connect two SAS sessions on different computers. This product allows you to connect to multiple remote SAS sessions, process applications, access data from a remote SAS session on your local machine, and enables you to combine SAS data from incompatible systems into one data set (SAS/CONNECT Software, Usage and Reference, Version 6, Second Edition).

SAS/CONNECT Software consists of two main services, *compute services* and *remote data services*. The compute services consist of remote session processing (RSUBMIT and RSPT), whereas the remote data services consist of data transfer (UPLOAD and DOWNLOAD) and remote library services (LIBNAME).

In a typical SAS/CONNECT session a link is established between two SAS sessions running on different hosts. Once the connection is established the user has access to the services and resources available to both sessions.

As a collaboration tool SAS/CONNECT can be extremely useful as it allows users to maximize a variety of system resources. Some of the more common reasons for choosing SAS/CONNECT software are:

- Users may have access to other SAS licensed products (e.g. SAS/STAT®, SAS/GRAPH®) on a remote system that is not available on their desktop system.
- The amount of data may be too large to be moved and/or the processing can be completed more efficiently on the remote system than on the desktop.
- The data is updated too frequently on the remote system to continuously transfer to the desktop for analysis.
- Multiple SAS files can be transferred in a single step.
- SAS files can be seamlessly moved between releases or across operating systems.

(Note: the reader is referred to SAS/CONNECT Software, Usage and Reference for a complete explanation of this product.)

Another client-server solution from SAS Institute is SAS/SHARE software. This product is often employed when multiple users need to update the same SAS data libraries or SAS files concurrently. Generally, the file systems on most operating systems prevent multiple users from accessing or making changes to the same file. However, through the use of SAS/SHARE software, the same libraries or files can be modified while ensuring that inconsistent changes are not made or important new information is not overwritten (Aster, 1994).

Several advantages to SAS/SHARE software have been noted by Aster (1994):

- Multiple users can modify or update separate observations in a SAS data set or catalog.
- The software works without modifying current SAS programs.
- SAS/CONNECT and SAS/ACCESS® software can be employed in combination with SAS/SHARE software.
- SAS/SHARE allows file sharing between the MVS and CMS operating systems.

Of course the disadvantage to both client-server solutions is that they are separate SAS modules, requiring an additional purchase and maintenance on each client machine.

THE WORLD WIDE WEB

One would be hard pressed today to find a computer enthusiast in business, government, or education who is unaware of the World Wide Web.

Although the internet has been in existence since 1962, most individuals really became aware of the internet and the concept of the WWW in 1993 with the birth of Mosaic, the first Web browser. Today it is estimated that there are over a million Web servers with more and more individuals and organizations adding information daily. (For a complete history of the internet the reader is referred to the Hobbes' Internet Time Line, http://info.isoc.org/guest/zakon/Internet/History/HIT.html.)

Clearly the Web has become the cutting edge technology for information delivery. As we learn more about the Web and its applications, more and more individuals are moving from merely being Web "surfers" (those who read and search Web pages) to actual Web information providers.

SAS OUTPUT AND THE WEB

A Web page is written using HyperText Markup Language (HTML). HTML is a document-layout and hyperlink-specification language. Within each HTML document are text and embedded directions called *tags*, which are codes surrounded by the < > symbols. These tags are not displayed by Web browsers, but define the layout and syntax of the document. (Musciano & Kennedy, 1996).

Although there are many standard HTML tags, only a handful are needed to write a simple HTML document:

```
<html>
<head>
<title> A Simple HTML Document </title>
<body>
<h1>A NESUG97 Example</h1>
This is a paragraph.
</body>
</html>
```

One of the simplest ways to deliver static SAS output on the Web is to utilize the <pre> tag.

The <pre> or preformatted text tag causes the browser to render text in the exact character and line spacing defined in the source document so that extraneous leading and trailing spaces are recognized (Musciano & Kennedy, 1996). This tag will display the text between the <pre> and </pre> tags in a monospaced font and is very useful for displaying columns of text or numbers as in PROC CONTENTS or PROC MEANS, or retaining the integrity of graphics output such as PROC CHART or PROC PLOT.

Although the necessary HTML tags can be added manually to any static SAS log or output, it is useful to employ a programming method for serving large numbers of static SAS pages. Larry Hoyle (1997) provides SAS users with a wonderful overview of HTML and step by step instructions on how simple HTML tags can be easily incorporated into SAS output utilizing PUT statements and redirecting SAS output with the PRINTTO procedure.

The following program has been adapted for the MVS system from the code developed by Hoyle (1997):

```
//  (JOB statement)
//  JCLLIB ORDER=(ZABCRUN.PROCLIB)
//  EXEC SAS
//SYSIN DD *
libname IN SPSS 'aaaaiii.spssname';
filename TESTHTML
   'aaaaiii.nesug97.contents.html'
   DISP=(NEW,CATLG) UNIT=FILE
   SPACE=(TRK,(1,1)) RECFM=FB
   LRECL=80 BLKSIZE=8000;
options ls=64 nodate pageno=1;

data _null_;
file TESTHTML;
put '<html>';
put '<head>';
put '<title>HTML for NESUG97</title>';
put '</head>';
put '<body>';
put '<pre>';

*redirect PROC output to the HTML file;
proc printto print= TESTHTML;
title 'Contents from SPSS system file';
proc contents data=IN._FIRST_;

proc printto;

data _null_;
file TESTHTML mod;
put '</pre>';
put '</body>';
put '</html>';
endsas;
```

In the example above, a html source file called ‘nesug97.contents.html’ is created on the MVS system. PUT statements write the standard tags (<html>, <title>, <head>, <body>, and <pre>) into the empty file. PROC PRINTTO redirects the CONTENTS procedure output to the new html source file.

Output 2 in the Appendix illustrates the final html source code generated by the program above. The previously discussed PROC CONTENTS output could then be easily displayed by a Web browser without losing the spacing and format structure of the MVS display.

SAS INSTITUTE WEB TECHNOLOGIES

SAS Institute is currently developing a full range of Web solutions that allow users to effortlessly display static SAS output, as well as tools that provide dynamic information delivery (for detailed information about these solutions, see the SAS Institute Research and Development Web Tools index page at http://www.sas.com/rnd/web/sitemap.html).

Although many SAS Web development tools are currently available for beta testing, at the time of this writing, most are only accessible by PC and UNIX operating systems. However, with the next MVS enhanced maintenance release (6.09E), several of these Web tools (HTML Formatting Tools and the Application Server) will soon become available[2].

STATIC INFORMATION DELIVERY

The Web is an extremely efficient vehicle for delivering static statistical data. Batch reports or graphs that are continuously updated and distributed in paper form is a very expensive method of distributing information to a large number of employees. By utilizing a Web browser to view such information, distribution costs remain minimal while critical and time sensitive information can be served more efficiently.

The majority of the information that is currently displayed on the WWW is static data. Static data can be text or graphic information that is displayed from a Web browser and usually requires a webmaster to update the pages as new information becomes available.

SAS Institute has several Web tools under development that can assist in the delivery of static information. These tools are known collectively as SAS HTML Formatting Tools (see SAS Research and Development, http://www.sas.com/rnd/web/publish.html, for more details).

There are three main types of SAS HTML Formatting Tools:

- **HTML Output Formatter** - saves output from any SAS procedure to an HTML file.
- **HTML Data Set Formatter** - converts SAS data sets to HTML 3.x tables.
- **HTML Tabulate Formatter** - Converts Proc TABULATE output into HTML 3.x tables.

No knowledge of HTML is required to use the SAS HTML Formatting Tools. These Web solutions allow you to easily create SAS output with valid HTML tags (not just output surrounded by the <pre> tag as previously described). Each of the formatting tools above is written as a SAS macro which defines a set of properties to format the text of the output. Either the default properties can be used or users can customize their own properties.

DYNAMIC INFORMATION DELIVERY

As more and more users become Web savvy, the demand for data delivery is moving from static Web pages to pages which allow interactivity between the user and the server, such as querying databases or running processes from a Web browser. SAS Institute is currently developing a number of solutions which allow users to create Web content on demand (See SAS Research and Development, http://www.sas.com/rnd/web/IntrNet.html for a complete description).

One of the components of the SAS/IntrNet™ Compute Services is the Application Dispatcher. The Application Dispatcher consists of:

- **An HTML Formatted Web Page** - may contain a HTML form, a hypertext link to the Application Broker, an inline image, or an object (such as a Java applet, ActiveX control, or Plug-in), that sends an immediate request to the Application Broker.
- **Application Broker** - a CGI program that resides on a Web server and interprets a form request and passes it to the Application Server.
- **Application Server** - A SAS session that receives and processes information from the Application Broker. Results are sent back to the Broker, which in turn, sends the results back to the user's Web browser.
- **Dispatcher Applications** - SAS code (a program, source entry, SCL entry, or compiled macro) which accepts the form information that is passed from the Application Broker.

(See SAS Research and Development, http://www.sas.com/rnd/web/dispatch/index.html, for more details).

At the time of this writing, SAS Institute is planning to ship the Application Server component of the Application Dispatcher for MVS in the Summer of 1997. SAS Institute is currently researching a version of the Application Broker for MVS, however, this version is still under development and will probably not ship at the same time as the Server component for MVS. Users will have to implement the Broker component on a UNIX or Windows server until the MVS version of the component is available[3].

NIH SILK WEB SERVICES

Recognizing the need for Web access to MVS legacy data, the NIH Computer Center has recently implemented a new service, **S** ecure **I**nternet-**L**in**K**ed (SILK) Web Technology. Typically in order to gain Web access to MVS data, NIH users had to download their output to another platform. This new system allows immediate Web access to virtually any type of MVS data. The data may be any type supported by Web browsers (e.g. text, HTML,GIF, JPG) (NIH/DCRT Interface 198, 1996a).

Users can choose to serve their data from either a public server (accessible by anyone), **http://silk.nih.gov/public/**, or a secure server that takes advantage of the MVS RACF facility, **http://silk.nih.gov/secure/**. Allocating output or data for Web access is as simple as saving the file on MVS with the following syntax:

aaaaiii.@www.name

Files saved with RACF protection, which restricts access to specific individuals or groups, access their files via the secure server, whereas, all unprotected files are accessed by way of the public server (NIH/DCRT Interface 198, 1996b).

The advantages to this new facility as described in the NIH SILK Web Seminar[4] are:

- MVS output or data can be served directly from the MVS system without the need to download to another platform.
- The MVS system provides Secure Socket Layer (SSL) encryption and the native MVS architecture is resistent to hackers.
- Solves many maintenance problems of the traditional client-server solutions.
- Employs an accepted user interface (Web browser), requiring a minimal learning curve for users.

NETSCAPE BROWSER SUITES

Netscape Communications Corporation is currently marketing an integrated suite of products which will allow users to more effectively access and manage information. Netscape Communicator (Professional Edition) is a new product composed of eight integrated components:

- **Netscape Navigator** - updated and revised premier Web browser.
- **Netscape Messenger** - open standards-based mail component.
- **Netscape Collabra** - provides internet newsgroup access, can navigate and create discussion groups.
- **Netscape Composer** - HTML authoring tool.
- **Netscape Conference** - real time audio and data collaboration component.
- **Netscape Calendar** - cross-platform calendar and scheduling component.
- **Netscape AutoAdmin** - simplifies administration and maintenance of Netscape products throughout the enterprise.
- **Netscape IBM® Host On-Demand** - 3270 legacy access component.

(For a thorough discussion of products and components, see Netscape Communications Corporation, http://home.netscape.com/comprod/products /communicator/guide.html.)

As Web access becomes more inherent to our daily work environments, Netscape Communications Corporation has recognized the usefulness of a suite of products that provides users with greater data access from multiple resources, all through the simplicity of their browser interface.

Of interest to MVS users will be the Netscape IBM Host On-Demand component which allows Web access to legacy data (e.g. MVS, VM, PROFS, DB2®). This component (written by IBM in Java) provides terminal access to any IBM or compatible 3270 host system. Configuring a session simply requires a host name and a port number inserted into the Java applet. Once connected you will see legacy information just as though you were viewing it from a dummy terminal.

Some of the features of this new component are:

- TN3270 host emulation.
- Multiple 3270 sessions can be run concurrently.
- Window sessions are resizeable with dynamically adjustable fonts.
- Includes mapped 3270 keyboard function keys.
- Full HTML help.

(Netscape Communications Corporation, http://home.netscape.com/comprod/products /communicator/guide.html#hostapp.)

CONCLUSION

In today’s research environments, users have many options for accessing and sharing legacy data, making cross-platform collaborations less daunting. Collaboration choices are no longer restricted to merely sending raw data files and their associated programs to colleagues.

Today, we have many available collaboration techniques which can streamline our work and get new information to colleagues even faster and more efficiently than before. The explosive growth of the Web has changed the way data is delivered, and has led to a flurry of new products and solutions, each designed to make legacy data access easier.

NOTES

[1] NIH statistics on legacy data presented at the NIH Division of Computer Research and Technology, SILK Web seminar, March 20, 1997.

[2] Personal communication, Bryan Wolfe, SAS Institute, Inc., May 20, 1997.

[3] Personal communication, Bryan Wolfe, SAS Institute, Inc., May 20, 1997.

[4] SILK Web advantages presented at the NIH Division of Computer Research and Technology, SILK Web seminar, March 20, 1997.

ACKNOWLEDGEMENTS

Loring Ingraham, Ph.D./ Psychologist/ NIH/NIMH
SAS Technical Support/SAS Institute Inc./ Cary, NC

REFERENCES

Aster, R. (1994) *SAS Foundations From Installation to Operation*, New York: Windcrest/McGraw-Hill.

Calvert W.S., Ma J.M. (1996) *Concepts and Case Studies in Data Management*, Cary, NC: SAS Institute Inc.

Currall J. (1995) “Moving Statistical Data Between Packages”, *User Note: University of Glasgow Computing Service*, http://www.gla.ac.uk/CompServ/Doc/General/un33.html.

Hoyle L. (1997) “Interfacing the SAS System with the World Wide Web,” *Proceedings of the Twenty-Second Annual SAS Users Group International Conference,* 22, 723-732.

Kuligowski A.T., Roberts N. (1997) “From There to Here: Getting Your Data into the SAS System”, *Proceedings of the Twenty-Second Annual SAS Users Group International Conference,* 22, 311-320.

Mallus, T.T. (1996) “Quality Assurance and the SAS System”, *Proceedings of the Ninth Annual NorthEast SAS Users Group Conference,* 9, 556-565.

Musciano, C., Kennedy, B. (1996) *HTML The Definitive Guide*, Sebastopol, CA: O’Reilly & Associates, Inc.

NIH/DCRT (1996a) “A New Web Service - Smooth as SILK”, *Interface* , 198, 5-7.

NIH/DCRT (1996b) “Guidelines for Using the New SILK Web Facility”, *Interface* , 198, 28-30.

SAS Institute Inc. (1996) *SAS Companion for the MVS Environment, Version 6, Second Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1994) *SAS Connect Software, Usage and Reference, Version 6, Second Edition,* Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990) *SAS Language: Reference, Version 6, First Edition,* Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990) *SAS Procedures Guide, Version 6, Third Edition,* Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1989) *SAS Technical Report P-195, Transporting SAS Files between Host Systems*, Cary, NC: SAS Institute Inc.

TRADEMARKS

THE AUTHOR MAY BE CONTACTED AT:

National Institute of Mental Health
Laboratory of Brain and Cognition
Building 10, Room 4C104
10 CENTER DR MSC 1366
BETHESDA MD 20892-1366

Phone: (301) 435-4934
Fax: (301) 402-0921
E-mail: thalene_mallus@nih.gov

Appendix

Output 1. SPSS Interface Engine Output Example

```
NOTE: The initialization phase used 0.09 CPU seconds and 2839K.
 1          LIBNAME IN SPSS 'aaaaiii.spssname';
 NOTE: Libref IN was successfully assigned as follows:
       Engine:        SPSS
       Physical Name: aaaaiii.spssname
 2          PROC CONTENTS DATA=IN._FIRST_;
 3          ENDSAS;

 NOTE: The PROCEDURE CONTENTS printed pages 1-60.
 NOTE: The PROCEDURE CONTENTS used 0.42 CPU seconds and 4002K.

 NOTE: The SAS session used 0.53 CPU seconds and 4002K.
 NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
 The SAS System                         12:00 Wednesday, May 21, 1997

                              CONTENTS PROCEDURE

Data Set Name: IN._FIRST_                            Observations:         912
Member Type:   DATA                                  Variables:            3192
Engine:        SPSS                                  Indexes:              0
Created:       16:15 Thursday, July 14, 1994         Observation Length:   25536
Last Modified: 12:00 Wednesday, May 21, 1997         Deleted Observations: 0
Protection:                                          Compressed:           NO
Data Set Type: DATA                                  Sorted:               NO
Label:

                    -----Engine/Host Dependent Information-----

                 SPSSINFO: SPSS SYSTEM FILE.   IBM OS/MVS RELEASE 4.1
                 COMPRESS: YES
                 SPSSTYPE: SPSSX

               -----Alphabetic List of Variables and Attributes-----

  #    Variable    Type    Len      Pos    Format    Label
 ---------------------------------------------------------------------------------------------
1195    A1          Num       8     9515    8.2       IKW  FLAT AFFECT
1196    A2          Num       8     9523    8.2       IKW  DISTANT, IMPOVERISHED
1197    A3          Num       8     9531    8.2       IKW  INAPPROPRIATE
.... etc.
```

Output 2. SAS Generated HTML Source Code Example

```
<html>
<head>
<title>HTML for NESUG97</title>
</head>
<body>
<pre>
                 Contents from SPSS system file                1

                       CONTENTS PROCEDURE

Data Set Name: IN._FIRST_            Observations:         912
Member Type:   DATA                  Variables:            3192
Engine:        SPSS                  Indexes:              0
Created:       Thu, Jul 14, 94       Observation Length:   25536
Last Modified: 15:45 Mon, Jun 9, 97  Deleted Observations: 0
Protection:                          Compressed:           NO
Data Set Type: DATA                  Sorted:               NO
Label:

          -----Engine/Host Dependent Information-----

SPSSINFO: SPSS SYSTEM FILE.   IBM OS/MVS RELEASE 4.1

COMPRESS: YES
SPSSTYPE: SPSSX

      -----Alphabetic List of Variables and Attributes-----

   # Variable Type Len   Pos Format Label
----------------------------------------------------------------
1195 A1       Num    8  9515 8.2    IKW  FLAT AFFECT
1196 A2       Num    8  9523 8.2    IKW  DISTANT, IMPOVERISHED
1197 A3       Num    8  9531 8.2    IKW  INAPPROPRIATE
1198 A4       Num    8  9539 8.2    IKW  SILLY,UNUSUAL,BIZARRE
1199 A5       Num    8  9547 8.2    IKW  COLD
1200 A6       Num    8  9555 8.2    IKW  HOSTILE, ANGRY
1201 A7       Num    8  9563 8.2    IKW  ANGRY
1153 ABSNO    Num    8  9186 3.     KENDLER ABSTRACT NUMBER
1462 ABSTRCT  Num    8 11651 3.     MODIFIES ABSNO CASES W/

... more PROC CONTENTS output....

</pre>
</body>
</html>
```